

HTTP range requests

An HTTP [Range](#) request asks the server to send parts of a resource back to a client. Range requests are useful for various clients, including media players that support random access, data tools that require only part of a large file, and download managers that let users pause and resume a download.

Checking if a server supports partial requests

If an HTTP response includes the [Accept-Ranges](#) header with any value other than `none`, the server supports range requests. If responses omit the `Accept-Ranges` header, it indicates the server doesn't support partial requests. If range requests are not supported, applications can adapt to this condition; for instance, download managers can disable pause buttons that relied on range requests to resume a download.

To check if a server supports range requests, you can issue a [HEAD](#) request to inspect headers without requesting the resource in full. If you use [curl](#), you can use the `-I` flag to make a `HEAD` request:

BASH

```
curl -I https://i.imgur.com/z4d4kwk.jpg
```

This will produce the following HTTP request:

HTTP

```
HEAD /z4d4kwk.jpg HTTP/2
```

```
Host: i.imgur.com
```

```
User-Agent: curl/8.7.1
```

```
Accept: */*
```

The response only contains headers and doesn't include a response body:

HTTP

```
HTTP/2 200
content-type: image/jpeg
last-modified: Thu, 02 Feb 2017 11:15:53 GMT
...
accept-ranges: bytes
content-length: 146515
```

In this response, `Accept-Ranges: bytes` indicates that 'bytes' can be used as units to define a range (currently, no other unit is possible). The [Content-Length](#) header is also helpful as it indicates the total size of the image if you were to make the same request using the `GET` method instead.

Requesting a specific range from a server

If the server supports range requests, you can specify which part (or parts) of the document you want the server to return by including the [Range](#) header in a HTTP request.

Single part ranges

We can request a single range from a resource using `curl` for illustration. The `-H` option appends a header line to the request, which in this case is the `Range` header requesting the first 1024 bytes. The last option is `--output -` which will allow printing the binary output to the terminal:

BASH

```
curl https://i.imgur.com/z4d4kWk.jpg -i -H "Range: bytes=0-1023" --output -
```

The issued request looks like this:

HTTP

```
GET /z4d4kWk.jpg HTTP/2
Host: i.imgur.com
User-Agent: curl/8.7.1
Accept: */*
Range: bytes=0-1023
```

The server responds with a [206 Partial Content](#) status:

HTTP

HTTP/2 206

`content-type`: image/jpeg

`content-length`: 1024

`content-range`: bytes 0-1023/146515

...

(binary content)

The [Content-Length](#) header indicates the size of the requested range, not the full size of the image. The [Content-Range](#) response header indicates where this partial message belongs within the full resource.

Multipart ranges

The [Range](#) header also allows you to get multiple ranges at once in a multipart document. The ranges are separated by a comma.

BASH

```
curl http://www.example.com -i -H "Range: bytes=0-50, 100-150"
```

The server responds with the [206 Partial Content](#) status as shown below. The response contains a [Content-Type](#) header, indicating that a multipart byterange follows. The boundary string (3d6b6a416f9b5 in this case) separates the body parts, each of which has its own `Content-Type` and `Content-Range` fields:

HTTP

HTTP/1.1 206 Partial Content

`Content-Type`: multipart/byteranges; boundary=3d6b6a416f9b5

`Content-Length`: 282

--3d6b6a416f9b5

`Content-Type`: text/html

`Content-Range`: bytes 0-50/1270

<!doctype html>

<html lang="en-US">

```
<head>
  <title>Example Do
--3d6b6a416f9b5
Content-Type: text/html
Content-Range: bytes 100-150/1270

eta http-equiv="Content-type" content="text/html; c
--3d6b6a416f9b5--
```

Conditional range requests

When resuming to request more parts of a resource, you need to guarantee that the stored resource has not been modified since the last fragment has been received.

The [If-Range](#) HTTP request header makes a range request conditional: if the condition is fulfilled, the range request will be issued and the server sends back a [206](#) Partial Content answer with the appropriate body. If the condition is not fulfilled, the full resource is sent back, with a [200](#) OK status. This header can be used either with a [Last-Modified](#) validator, or with an [ETag](#), but not with both.

HTTP

[If-Range](#): Wed, 21 Oct 2015 07:28:00 GMT

Partial request responses

There are three relevant statuses, when working with range requests:

- A successful range request elicits a [206](#) Partial Content status from the server.
- A range request that is out of bounds will result in a [416](#) Requested Range Not Satisfiable status, meaning that none of the range values overlap the extent of the resource. For example, the first-byte-pos of every range might be greater than the resource length.
- If range requests are not supported, an [200](#) OK status is sent back and the entire response body is transmitted.

Comparison to chunked Transfer-Encoding

The [Transfer-Encoding](#) header allows chunked encoding, which is useful when larger amounts of data are sent to the client and the total size of the response is not known until the request has been fully processed. The server sends data to the client straight away without buffering the response or determining the exact length, which leads to improved latency. Range requests and chunking are compatible and can be used with or without each other.

See also

- Related status codes [200](#) , [206](#) , [416](#) .
- Related headers: [Accept-Ranges](#) , [Range](#) , [Content-Range](#) , [If-Range](#) , [Transfer-Encoding](#) .

Help improve MDN

Was this page helpful to you?

Yes

No

[Learn how to contribute.](#)

This page was last modified on Mar 13, 2025 by [MDN contributors](#).

